CLAIMS

1. A method for operating a multiple virtual processor system, the multiple virtual processor system including a program memory, a thread scheduling mechanism, and a physical processor, the method comprising:

storing a plurality of threads in the program memory, wherein a first thread of the plurality of threads comprises a plurality of first instructions including a YIELD instruction;

executing the first thread by systematically passing the first instructions from the program memory to the physical processor, and causing the physical processor to process the first instructions;

suspending execution of the first thread when the YIELD instruction is processed by the physical processor;

identifying a second thread from the plurality of threads for execution by the physical processor, wherein the second thread is selected by the thread scheduling mechanism based on a predefined schedule and the processed YIELD instruction; and

executing the second thread by systematically passing second instructions associated with the second thread from the program memory to the physical processor, and causing the physical processor to process the second instructions.

2. The method according to Claim 1, wherein the program memory comprises a volatile memory device, and wherein storing the plurality of threads comprises writing the plurality of threads from a non-volatile memory device into the program memory.

3. The method according to Claim 2, wherein the MVP system comprises a first discretely packaged semiconductor device, and

the non-volatile memory device comprises a second discretely packaged semiconductor device, and wherein writing the plurality of threads comprises transmitting data between the first and second discretely packaged semiconductor devices during operation of the MVP system.

4. The method according to Claim 1, wherein a portion of the program memory comprises a deterministic memory for continuously storing a pre-selected thread of the plurality of threads, and wherein storing the plurality of threads includes writing all instructions associated with the pre-selected thread into the deterministic memory during a system initialization period.

5. The method according to Claim 1, wherein the first thread includes operating state information that is loaded into the physical processor before executing the first thread.

6. The method according to Claim 1,

wherein executing the first thread comprises fetching the first instructions from the program memory using a first program counter, and

wherein executing the second thread comprises fetching the second instructions from the program memory using a second program counter.

7. The method according to Claim 1, wherein executing the first thread comprises selecting the first thread from the plurality of threads based on the predefined schedule.

8. The method according to Claim 1, further comprising:

suspending execution of the second thread based on the predefined schedule; and

resuming execution of the first thread.

9. The method according to Claim 1, wherein suspending execution of the first thread further comprises determining whether the second thread is available for execution.

10. A multiple virtual processor (MVP) system comprising:

a program memory for storing a plurality of threads; and

a processor core coupled to the program memory, the processor core including:

a thread scheduling mechanism for scheduling the execution of a first thread and a second thread based on a predetermined schedule,

a physical processor for processing instructions associated with a selected thread of the first and second threads, and

switching means for passing instructions associated with the selected thread from the program memory to the physical processor,

wherein the first thread includes a YIELD machine instruction,

wherein the processor core comprises means for notifying the thread scheduling mechanism when the YIELD machine instruction is processed by the physical processor during execution of the first thread, and

wherein the thread scheduling mechanism includes means for suspending execution of the first thread and for initiating execution of the second thread by the physical processor based on the predefined schedule and the processed YIELD instruction.

11. The MVP system according to Claim 10, wherein the program memory comprises a volatile memory device, and wherein storing the plurality of threads comprises writing the plurality of threads from a non-volatile memory device into the program memory.

12. The MVP system according to Claim 11, wherein the MVP system comprises a first discretely packaged semiconductor device, and the non-volatile memory device comprises a second discretely packaged semiconductor device.

13. The MVP system to Claim 10, wherein a portion of the program memory comprises a deterministic memory for continuously storing all instructions associated with a pre-selected thread of the plurality of threads.

14. The MVP system according to Claim 10, wherein the first thread includes operating state information that is loaded into the physical processor before executing the first thread.

15. The MVP system according to Claim 10, further comprising:

a first program counter for fetching the first instructions from the program memory during execution of the first thread; and

a second program counter for fetching second instructions associated with the second thread from the program memory during execution of the second thread.

16. The MVP system according to Claim 10, wherein the thread scheduling mechanism further comprises means for determining an availability of the second thread for execution by

the physical processor before initiating execution of the second thread.

17. A multiple virtual processor system including a program memory, a thread scheduling mechanism, and a physical processor, the multiple virtual processor system also comprising:

means for storing a plurality of threads in the program memory, the plurality of threads including a first thread comprising a plurality of first instructions including a YIELD instruction;

means for executing the first thread by systematically passing the first instructions from the program memory to the physical processor, and causing the physical processor to process the first instructions;

means for determining when the YIELD instruction is processed by the physical processor;

means for suspending execution of the first thread upon determining that the YIELD instruction has been processed by the physical processor; and

means for identifying and executing a second thread from the plurality of threads using the physical processor, wherein the second thread is selected based on a predefined schedule and the processed YIELD instruction.

18. The MVP system according to Claim 17, wherein the first thread includes operating state information, and wherein the MVP system further comprises means for loading the operating state information into the physical processor before executing the first thread.

19. The MVP system according to Claim 17, further comprising:

a first program counter for fetching the first instructions from the program memory during execution of the first thread; and

a second program counter for fetching second instructions associated with the second thread from the program memory during execution of the second thread.

20. The MVP system according to Claim 17, further comprising means for determining an availability of the second thread for execution by the physical processor before initiating execution of the second thread.